

BEYOND KNUTH'S NOTATION FOR UNIMAGINABLE NUMBERS WITHIN COMPUTATIONAL NUMBER THEORY

Antonino Leonardis, Gianfranco d'Atri and Fabio Caldarola

Received: 28 September 2020; Revised: 4 April 2021; Accepted: 14 April 2021

Communicated by Abdullah Harmancı

ABSTRACT. Literature considers under the name “unimaginable numbers” any positive integer going beyond any physical application. One of the most known methodologies to conceive such numbers is using *hyper-operations*, that is a sequence of binary functions defined recursively starting from the usual chain: addition - multiplication - exponentiation. The most important notations to represent such hyper-operations have been considered by Knuth, Goodstein, Ackermann and Conway as described in this work’s introduction. Within this work we will give an axiomatic setup for this topic, and then try to find on one hand other ways to represent unimaginable numbers, as well as on the other hand applications to computer science, where the algorithmic nature of representations and the increased computation capabilities of computers give the perfect field to develop further the topic, exploring some possibilities to effectively operate with such big numbers. In particular, we will give some axioms and generalizations for the up-arrow notation and, considering a representation via rooted trees of the hereditary base- n notation, we will determine in some cases an effective bound related to “Goodstein sequences” using Knuth’s notation. Finally, we will also analyze some methods to compare big numbers, proving specifically a theorem about approximation using scientific notation and a theorem on hyperoperation bounds for Steinhaus-Moser notation.

Mathematics Subject Classification (2020): 11A67, 11U99, 11Y99

Keywords: Computational number theory, unimaginable numbers, Knuth’s up-arrow notation, big data, number representation, Goodstein’s theorem

1. Introduction

Several methods and notations are been developed in the last century to work, or better to try to consider, very large numbers for which, in this paper, we propose the name of unimaginable numbers. One of the most known methodologies is the so-called Knuth up-arrow notation introduced by D.E. Knuth in 1976 (see

This work was carried out as part of the project *IoT&B “Internet of Things and Blockchain”*, CUP J48C17000230006, supported by POR Calabria FESR-FSE 2014-2020.

[19]) and strictly linked to the concept of hyper-operation and Ackermann function (see [1,23]). The idea of hyper-operation dates back to the early 1900s by A.A. Bennet (see [5]), and subsequently we re-find it in a group of Hilbert's students as W. Ackermann and G. Sudan. But the widespread contemporary names like tetration, pentation, hexation, or in general hyper- n operation were introduced by R.L. Goodstein in 1947 (see [17]) and gained popularity through Rudy Rucker's book *Infinity and the Mind* [24], published in 1982. Knuth up-arrow is not the only notation used today for very large numbers; there are in fact many other ways to write hyper-operators, as we may recall among others:

- square bracket notation, box notation and superscripts and subscripts notation (see [21,22]);
- Nambiar's notation (see [23]).

Moreover we point out that there are also so enormous numbers that even Knuth's notation and the previous ones, are not sufficient to represent them. For this purpose J.H. Conway introduced a more powerful notation based on recursion, to write extremely large numbers. It is known as Conway's chained arrow notation (see for example [16]) and can be viewed as a generalization of Knuth's arrow notation: in fact, in the case of a length 2 sequence $a \rightarrow b \rightarrow n$, it is equivalent to $a \uparrow^n b$ Knuth's notation. Similarly, the *Bowers' operator*, also called the *Bowers' exploding array function* (see [6]), is a more powerful numeral system proposed by J. Bowers and published on the web in 2002, which generalizes hyper-operators.

The Steinhaus-Moser notation (see [26]) is another way to express by recursion very big numbers. It is in fact more intuitive (thus fitting well within educational purposes) in its definition than the hyper-operations, and for its recursion properties will be applied to find an effective bound for certain couples in Goodstein's theorem (see below).

A relevant link between unimaginable numbers and computer science is related with the so called arbitrary-precision arithmetic and blockchain tools, as one can use such huge numbers to handle machine-computed big data. This work arose indeed from a discussion between the authors (during the preparation of "The First Symposium of the International Pythagorean School - Da Pitagora a Schützenberger: numeri inimmaginabili"¹) about the use of *grossone*, a recent definition of an arithmetical infinity (see [25] or also [2,3,7,8,9,13,14]), in order to compute limits in a similar fashion to non-standard analysis.

¹"Inimmaginabili" is the Italian plural word for "unimaginable", and has been modified by using the fancy letter "i" in order to resemble Knuth's up-arrows.

We will start the paper by giving a complete axiomatic definition of hyper-operators, linking this to Knuth's and Goodstein's notations. We will define the notion of meta-algorithm in order to define precisely the idea behind "repeating" an operation. After that, we will define a graph-theory representation of numbers linked to Goodstein's theorem (see [18]), which has also a simple set-theory interpretation when considering base 2, called rooted tree representation, and we will determine in some cases an explicit recursive algorithm for the number of steps required to reach zero for the so called "Goodstein sequences", as well as an effective bound for this number using Knuth's notation. We will conclude this work by applying various methods, among others from continued fractions (see [15]), to compare unimaginable numbers.

2. Extending Knuth's up-arrow notation

2.1. Historical notes. The basic arithmetical operations are defined recursively starting from the successor operation. The exponentiation, for instance, is a repeated multiplication. Knuth and Goodstein (see [19] and [17]) have further extended this definition, so that for example the tetration is a repeated exponentiation.

2.2. Arrow function definition. The work from Knuth and Goodstein can be formalized by the following general arrow-function:

- (1) $\uparrow(A, B, 0) := AB$;
- (2) $\uparrow(A, 0, k) := 1$ for $k \geq 1$;
- (3) $\uparrow(A, B + 1, k) := \uparrow(A, \uparrow(A, B, k), k - 1)$.

We can add in the mix also the following cases (satisfying recurrence law 3 as well):

- $\uparrow(A, B, -2) := A \bowtie B := \max(A, B) + 1$;
- $\uparrow(A, B, -1) := A + B$;
- $\uparrow(A, -1, k) := 0$ for $k \geq 2$.

This is a slightly modified version of the original one from Goodstein, which is related by the simple equality:

$$G(k, A, B) = \uparrow(A, B, k - 2)$$

and Knuth's notation is as well very similar, writing:

$$\begin{array}{ll} A \uparrow B := \uparrow(A, B, 1) & \text{[Normal exponentiation];} \\ A \uparrow\uparrow B := \uparrow(A, B, 2) & \text{[Tetration];} \\ A \uparrow\uparrow\uparrow B := \uparrow(A, B, 3) & \text{[Pentation];} \\ A \uparrow^k B := \uparrow(A, B, k) & \text{[Hyper-}(k+2)\text{ operation].} \end{array}$$

The last one is a compact expression for $A \uparrow \dots \uparrow B$ where A and B are separated by exactly k arrows. One could also use the symbol $\hat{}$ instead of each up-arrow, reobtaining the usual notation for exponentiation.

Important remark: after the normal multiplication, all the operations we have defined are **no more** commutative nor associative, and priority is to compute them all in order from right to left (*right associativity*).

Example 2.1. Let's compute the following tetration:

$$3 \uparrow\uparrow 4 = 3 \uparrow 3 \uparrow 3 \uparrow 3 = 3 \uparrow 3 \uparrow 27 = 3 \uparrow 7625597484987$$

which is a number with exactly 3638334640025 digits.

Example 2.2. Let's compute the following pentation:

$$2 \uparrow\uparrow\uparrow 3 = 2 \uparrow\uparrow 2 \uparrow\uparrow 2 = 2 \uparrow\uparrow 2 \uparrow 2 = 2 \uparrow\uparrow 4 = 2 \uparrow 2 \uparrow 2 \uparrow 2 = 2^{16} = 65536$$

which is, for instance, the number of characters which can be stored in a 2-byte system on a computer.

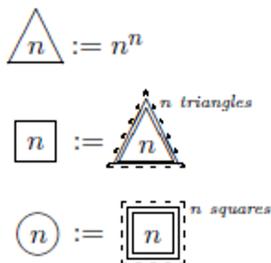
Remark 2.3 (Trivial towers). The following equalities hold for any $k \geq 1$:

$$\begin{aligned} \forall x \in \mathbb{N} \quad 1 \uparrow^k x &= x \uparrow^k 0 = 1 \\ \forall x \in \mathbb{N} \quad x \uparrow^k 1 &= x \\ 2 \uparrow^k 2 &= 4. \end{aligned}$$

For new density results on Knuth's powers see [12], or see also [10,11] for some links between unimaginable numbers, gross-one and new algebraic and geometric constructs arising from Fibonacci numbers.

2.3. Steinhaus-Moser notation. See [26] for the original definition.

Definition 2.4. Steinhaus-Moser notation uses geometrical shapes to express big numbers. A number surrounded by a shape will have the following meaning:



Using a more functional notation, we will define (f^n means we compose f with itself n times):

- $\Delta(n) := n^n$;
- $\square(n) := \Delta^n(n)$;
- $\circ(n) := \square^n(n)$.

One could also use a regular pentagon instead of the circle and continue the sequence for any regular k -gon; we will denote the generalized Steinhaus-Moser notation using the recursive function:

$$\begin{aligned} SM_3(n) &:= n^n = \Delta(n), \\ SM_{k+1}(n) &:= SM_k^n(n). \end{aligned}$$

Example 2.5. The number **Mega** is defined as $\textcircled{2}$, that is:

$$\circ(2) = \square(\square(2)) = \square(\Delta(\Delta(2))) = \square(\Delta(4)) = \square(256) = \Delta^{256}(256)$$

where the last expression contains already too many triangles to be computed explicitly.

Example 2.6. Another important number expressed with this notation is the **Megiston**, defined as $\textcircled{10}$.

2.4. Meta-algorithms. All operations we have considered give an “algorithm” to compute a natural number; we may construct a “meta-algorithm” by considering a string where the instances of “ $\overbrace{\dots}^k$ ” mean we should repeat the dotted part k times; for instance:

$$\overbrace{2 \uparrow 5}^3$$

means to construct the algorithm $2 \uparrow 2 \uparrow 2 \uparrow 5$, that is $2^{2^{2^5}}$.

We write the meta-function “EXPAND” meaning the bracketed string should be expanded with the rule just mentioned. We can now define a “generalized arrow function” as:

$$\uparrow(A, B, k, C) := \text{EXPAND} \left[\overbrace{A \uparrow^{k-1} C}^B \right]$$

so for instance we have the previous “generalized tetration”: $\uparrow(2, 3, 2, 5) = 2^{2^{2^5}}$.

In general $\uparrow(A, B, k) = \uparrow(A, B, k, 1)$, so it is indeed a generalization of the previous definition.

3. Rooted tree representation

3.1. Binary case. We consider the set T containing the following elements:

- $\emptyset \in T$.
- A finite set of elements of T ($A = \{a_i \in T\}_{i \in I}$) is itself an element of T ($A \in T$), and vice-versa any element of T contains only elements of T without infinite descending chains.

This set has the following properties:

- Any element $t \in T$ can be associated to a rooted tree: one recursively builds the tree for each element of t , and then connects their roots to a new root for t itself. Thus tree is also *unredundant*, in the sense that different branches of the same node are distinct (from the fact that elements in a set are all different from each other). By this definition, the tree associated to the empty set will be a root with no branches.
- It is defined a *height* function $H : T \rightarrow \mathbb{N}$ as:

$$H(\emptyset) := 0; \quad H(A) := 1 + \max_{t \in A} H(t)$$

which is well defined from the assumption on descending chains.

- There is an algorithmic bijection $f : T \xrightarrow{\cong} \mathbb{N}$ defined recursively as follows:
 - $f(\emptyset) = 0$;
 - $f(A) = \sum_{t \in A} 2^{f(t)}$.

Before going further we briefly prove bijectivity. Indeed, we must prove that $f(A) = f(B) \rightarrow A = B$, and we will proceed by induction on $\max(H(A), H(B))$. We suppose inductively that $f(a) = f(b) \rightarrow a = b$ is true for $\max(H(a), H(b)) < \max(H(A), H(B))$. By the uniqueness of the binary expansion for natural numbers, $f(A)$ and $f(B)$ have the same non-zero digits, which correspond to elements $a \in A, b \in B$ where $f(a)$ and $f(b)$ give the position of the digit. For each such couple we must have $f(a) = f(b)$ and by the inductive assumption we deduce $a = b$, so that A and B must have the same elements QED.

Using this bijection we are authorized from now on to not distinguish between A and $f(A)$. We define:

$$M_k := \max\{A \in T \mid H(A) = k\},$$

$$m_k := \min\{A \in T \mid H(A) = k\}.$$

The first one is obtained when A contains all possible elements t of height $< k$. Thus:

- $M_0 = 0$;
- $M_1 = 2^0 = 2^1 - 1 = 1$;
- $M_2 = 2^0 + 2^1 = 2^2 - 1 = 3$;
- $M_3 = 2^0 + 2^1 + 2^2 + 2^3 = 2^4 - 1 = 15$;
- $M_4 = 2^0 + \dots + 2^{15} = 2^{16} - 1$.

The second one is instead obtained by the recursion $m_0 = \emptyset$; $m_k = \{m_{k-1}\}$. Considering the recursive sequence

- $a_0 = 0$,
- $a_{i+1} = 2^{a_i}$,

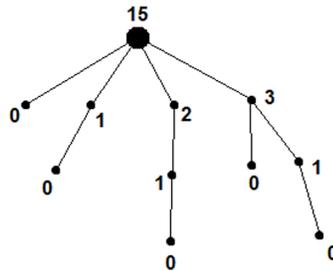
one can immediately prove by induction that $M_k = a_{k+1} - 1$ and $m_k = a_k$, so that height is proven to be a non-decreasing function. Using Knuth's up-arrow notation, we have $m_k = 2 \uparrow\uparrow (k - 1)$ so that every element of T is found in a specific interval depending on its height:

$$2 \uparrow\uparrow (H(A) - 1) \leq A < 2 \uparrow\uparrow H(A).$$

Example 3.1. $M_3 = \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}$ is the set of all elements with height at most 2, so it is the greatest one with height 3, and indeed it satisfies:

$$f(M_3) = 2^0 + 2^1 + 2^2 + 2^3 = 1 + 2 + 4 + 8 = 15.$$

The associated rooted tree is the following (we write on each node the integer corresponding to its height):



Remark 3.2. With the usual notation $\mathcal{P}(A) := \{X \subset A\}$, we notice that for any $k \geq 1$ the following facts hold:

- $M_k = \mathcal{P}(M_{k-1})$;
- $\#(M_k) = m_k$, because M_k contains all numbers from 0 to $m_k - 1$.

Summing up those results, we have that the tetration $2 \uparrow\uparrow (k-1) = m_k$ represents exactly the cardinality of the set

$$\mathcal{P}^k(\emptyset) := EXPAND \left[\overbrace{\mathcal{P}(\emptyset)}^k \overbrace{)}^k \right].$$

More generally, the “generalized tetration” gives the cardinality of the nested power set

$$\#(\mathcal{P}^k(A)) = \uparrow(2, k, 2, \#(A)).$$

For example, $\#(\mathcal{P}^k(\emptyset)) = \uparrow(2, k, 2, 0) = \uparrow(2, k-1, 2) = 2 \uparrow\uparrow (k-1)$.

3.1.1. Comparison. Comparing two elements $A, B \in T$ is performed with the following rule: one recursively can compare elements of $A \Delta B$ (symmetric difference), and put them in order; if its biggest element comes from A , then A is the bigger number, otherwise B is the bigger one.

Remark 3.3. For this purpose, and other following purposes, we remind that (as we are talking about sets) the order in theory doesn't matter, but actually we should consider every set as being already ordered so that finding the biggest element becomes an easy task.

3.1.2. Successor. We want to compute $s(A)$ for some $A \in T$. If $A = M_k$ for some k , then one has to consider directly $s(A) = m_{k+1}$. Otherwise, let $n_A \neq A$ be the unique natural number such that

$$n_A \notin A \wedge h \in A \forall h < n_A,$$

which is distinct from A precisely because A is not an M_k . Then one just has to remove every h smaller than n_A from A and insert instead the element n_A .

3.1.3. Addition. The sum of A and B is obtained by joining their elements; if an element t is repeated twice, one performs a carry and inserts **instead** the element $s(t)$, which could as well require another carry.

3.1.4. Multiplication. To multiply A and B one considers:

$$A \cdot B = \sum_{(a,b) \in A \times B} (c := \{a + b\})$$

which in usual representation would mean:

$$\left(\sum 2^a \right) \cdot \left(\sum 2^b \right) = \sum 2^{a+b}.$$

3.1.5. Generalized rooted tree representation. General case has been considered first by Goodstein (see [18]). When considering non-binary bases b the set representation fails to be useful, unless considering a more sophisticated notation. A representation will be a couple (b, s) with $b \in \mathbb{Z}_{\geq 2}$ and s a string in the language $\{1, 2, \dots, b-1, "+", "(", ")"\}$. The string will be interpreted as between digits and brackets there were the full expression " $\cdot b \uparrow$ ". For instance, if $b = 3$,

$$2(1() + 2(1())) + 1(2(())) = 2 \cdot 3 \uparrow (1 + 2 \cdot 3 \uparrow (1) + 1 \cdot 3 \uparrow (2)) = 2 \cdot 3^{16}.$$

More formally, after fixing the base b , one considers the following type of strings:

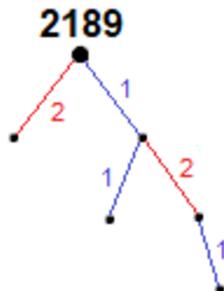
- **EMPTY**: an empty string **representing** 0;
- **SUM**: any number of **DIGIT** strings (see below) separated by the usual "+" symbol and having different exponents, **representing** the sum of **values** of the **DIGIT** components;
- **MISC**: an **EMPTY** or **SUM** string;
- **DIGIT**: a digit $0 \leq d < b$ followed by a **MISC** string representing some number s (called "exponent") into brackets, which has **value** $d \cdot b^s$.

The final string m has the **MISC** form, and is associated to a uniquely determined value in \mathbb{N} (precisely the number represented by m). This kind of approach is typical of computer science definitions for metadata (see for example [4]).

Remark 3.4. We recall that again **order doesn't matter** in **SUM** strings, as that's the reason we keep using plus symbol as a separator, but for computational purposes one should always consider sums ordered by digits' exponents.

We also may consider again rooted trees, where now connections between nodes are **labeled** with a digit from 1 to $b-1$.

Example 3.5. If $b = 3$, using as labels the colors **blue**=1 and **red**=2, we have the following representation:



where the bracketed algorithm is:

$$\begin{aligned} 2() + 1(1()) + 2(1()) &= 2 \cdot 3^0 + 1 \cdot 3^{1 \cdot 3^0 + 2 \cdot 3^{1 \cdot 3^0}} = \\ &= 2 + 3^{1+6} = 2 + 2187 = 2189. \end{aligned}$$

We notice that also in this case we can define the **height** of a graph, and that the sequences of minimum/maximum elements with a certain height can be found as well:

$$b \uparrow\uparrow (H(A) - 1) \leq A < b \uparrow\uparrow H(A)$$

because the minimum $m_k = b \uparrow\uparrow (H(A) - 1)$ is attained when there is a single path of digits 1 while the maximum is the sum of terms $(b-1) \cdot b^k$ with $k < b \uparrow\uparrow (H(A) - 1)$, a geometric progression having indeed sum $M_k := [b \uparrow\uparrow H(A)] - 1$.

3.1.6. Goodstein's theorem. Goodstein's theorem (see [18]) has an interesting interpretation within the topic of rooted tree notation. We recall that Goodstein's theorem involves the function which, given a couple (b, A) of a base $b \in \mathbb{N}$ and a rooted tree in that base, can be interpreted as:

$$F(b, A) = (b + 1, A - 1)$$

where the tree A is reread in the new base $b + 1$ and then decreased by 1.

Goodstein's theorem says that iterating this function one definitely stops at the value 0 whatever is the first element to which it is applied, and even though the function increases dramatically for almost every element. The proof relies on substituting every basis with the ordinal ω , so that the values obtained by this iteration form a strictly decreasing succession of ordinals for which we know it must stop somewhere, and the only possibility is 0. The rooted tree representation makes clear why the function is decreasing, as any natural number involved in representation is less than ω in the theory of ordinals.

We also point out that reinterpreting the proof using rooted trees doesn't actually require ordinal theory: geometrical properties of rooted trees should be enough to prove the assert without even involving the base, and this could indeed be studied in a more detailed future work on the topic.

We conclude this section by calculating an effective bound for some Goodstein sequences. For any integer a we set $\check{a} := a - 1$.

Theorem 3.6. *Let $b > 1$ and $k > 0$ be integers. We denote by $B_k(b)$ ($k < b$) the base required for the couple $(b, \check{\check{k}} + \dots + \check{\check{1}} + \check{\check{()}})$ to reach the stopping value*

–1. Then we have an explicit recursion to describe this function:

$$\begin{aligned} B_1(b) &= 2 \cdot b \\ B_k(b) &= B_{k-1}^b(b) \end{aligned}$$

where the latter exponent means one should repeatedly apply b times the function B_{k-1} . For example

$$B_2(b) = \text{EXPAND} \left[\overbrace{2 \cdot}^b b \right] = 2^b b.$$

Corollary 3.7. *If A is a tree in the base $b > 2$ with height $H(A) \leq 2$, then Goodstein's algorithm applied to the couple (b, A) reaches the stopping point $(B, -1)$ when:*

$$B = B_b(b) < SM_{b+1}(b) \leq (2b - 2) \uparrow^{b-1} (b + 1)$$

where SM_k is the generalized k -gon Steinhaus-Moser function (see Definition 2.4) and the last inequality comes from Theorem 4.9 proved below.

We remark that this corollary tells us that $B - b - 1$ is an effective bound for the algorithm to reach 0.

Proof of Theorem 3.6. The first equality comes from the fact that every step decreases the only digit by 1 while increasing the basis by the same amount; thus going from the digit $b - 1$ to -1 requires b steps, which increase the basis from b to $2b$. The second one derives from the fact that every time the biggest digit decreases by 1, the other $k - 1$ digits come from the same problem where the basis is updated by applying the function B_{k-1} , and this has to be done b times. \square

Proof of Corollary 3.7. To prove the corollary, it is known that it is enough to do it for $A = \check{b}(\check{b}) + \dots + \check{b}(1) + \check{b}() = m_2$, and we notice that in this case $B_2(b) = 2^b b < 3^b \leq b^b = \Delta(b)$ so that the recursive definition forces $B_k(b) < SM_{k+1}(b)$ (compare Definition 2.4) and $B = B_b(b) < SM_{b+1}(b)$ as wanted. \square

We close this section by pointing out to the reader that it is possible to write a closed formula for $B_k(b)$ valid for any $b \geq 2$ and $k \geq 1$:

$$B_k(b) = b \cdot 2 \uparrow b \uparrow (k - 1).$$

4. Comparing big numbers

4.1. Continued fractions preliminaries.

Lemma 4.1 (Dirichlet property). *A continued fraction approximant $\frac{b}{a}$ to an irrational number $x > 1$ satisfies:*

$$\left| \frac{b}{a} - x \right| < \frac{1}{ab} < \frac{1}{a^2}.$$

Proof. It is well known that x is between $\frac{a}{b}$ and the next approximant $\frac{c}{b}$, and that:

$$\left| \frac{b}{a} - \frac{c}{b} \right| < \frac{1}{ab} < \frac{1}{a^2}$$

so that the assertion follows immediately. \square

Lemma 4.2. *Given $A < B \in \mathbb{N}$ such that $x = \frac{\ln B}{\ln A}$ is an irrational real number, the continued fraction approximants $\frac{b}{a}$ to x are such that:*

$$e^{-\epsilon} < \frac{A^b}{B^a} < e^\epsilon$$

where $\epsilon := \frac{\ln A}{b}$.

Proof. By Lemma 4.1 we have:

$$\begin{aligned} -\frac{1}{a^2} &< \frac{b}{a} - \frac{\ln B}{\ln A} < \frac{1}{a^2} \\ -\frac{1}{b} &< b - \frac{\ln B}{\ln A} a < \frac{1}{b} \\ A^{-\frac{1}{b}} &< \frac{A^b}{B^a} < A^{\frac{1}{b}} \end{aligned}$$

and we conclude observing that $e^{\pm\epsilon} = A^{\pm\frac{1}{b}}$ by definition of ϵ . \square

4.2. Undistinguishable numbers. See also the introduction to [20].

Theorem 4.3. *If A, B, a, b, x are as in Lemma 4.2 and $k > 1$ is a natural number, then A^b and B^a are k - or $(k+1)$ -undistinguishable powers when:*

$$b > \ln A \cdot 2 \cdot 10^{k+1}$$

in the sense that in scientific notation they have the same expression considering only the first k or $k+1$ significant digits of their decimal expansion.

Proof. Two numbers whose ratio is bounded by the number $\frac{1}{1-0.5 \cdot 10^{-(k+1)}} \approx 1 + 0.5 \cdot 10^{-(k+1)} \approx \exp(0.5 \cdot 10^{-(k+1)})$ are sure to have the same scientific notation expression to the $(k+1)$ -th significant digit, possibly differing for the last one (including the possibility of a carry); in this case the $(k+1)$ -th digit must be the

same (the difference between the two approximations is bigger than double the difference of the two numbers) and we will have the same approximation to the k -th digit instead.

Now we can apply Lemma 4.2, where by hypothesis $\epsilon < 0.5 \cdot 10^{-(k+1)}$ so that the ratio A^b and B^a is bounded by e^ϵ , i.e. the number we just talked about, and we know already that in this case the thesis holds. \square

Example 4.4. For $A = 2$ and $B = 3$ we can consider the approximant:

$$\frac{b}{a} = [1; 1, 1, 2, 2, 3, 1, 5, 2, 23, 2, 2, 1, 1, 55] = \frac{16785921}{10590737}.$$

Being $b > \ln 2 \cdot 2 \cdot 10^7 \approx 13862944$, we know that $2^{16785921}$ and $3^{10590737}$ are 6-undistinguishable powers, and indeed both have the following expression in scientific notation:

$$\begin{aligned} 5.3191952\dots \cdot 10^{5053065} &\approx 5.31920 \cdot 10^{5053065} \\ 5.3191955\dots \cdot 10^{5053065} &\approx 5.31920 \cdot 10^{5053065} \end{aligned}$$

that is, they give the same approximation to the 6-th digit (one of them actually approximate to 5.319196 to the 7-th digit, so we must take one digit less for the exact correspondence).

4.3. Comparing Knuth and Steinhaus-Moser notation. We will consider only positive integers when not specified otherwise. Moreover k will be a counter ranging from 0 to n .

Proposition 4.5. *The square symbol is comparable to the tetration in the following way:*

$$n \uparrow\uparrow (n+1) \leq \square(n) \leq n \uparrow n \uparrow (n+1) \uparrow\uparrow (n-1) \leq n \uparrow\uparrow (n+2).$$

More precisely, a sequence of k triangles has the property:

$$n \uparrow\uparrow (k+1) \leq \Delta^k(n) \leq n \uparrow n \uparrow (n+1) \uparrow\uparrow (k-1).$$

Proof. The first inequality is straightforward, as we have by induction:

$$\Delta(n \uparrow\uparrow k) = [n \uparrow\uparrow k]^{n \uparrow\uparrow k} = n^{n^{\{[n \uparrow\uparrow (k-2)] + [n \uparrow\uparrow (k-1)]\}}} \geq n^{n^{[n \uparrow\uparrow (k-1)]}} = n \uparrow\uparrow (k+1)$$

so that n in k triangles is always $\geq n \uparrow\uparrow (k+1)$.

For the second inequality, we notice that:

$$\begin{aligned} \Delta \left(n^{n^{[(n+1) \uparrow\uparrow (k-2)]}} \right) &= \left\{ n^{n^{[(n+1) \uparrow\uparrow (k-2)]}} \right\}^{n^{n^{[(n+1) \uparrow\uparrow (k-2)]}}} \\ &= n^{n^{\{[(n+1) \uparrow\uparrow (k-2)] + [n \uparrow\uparrow (n+1) \uparrow\uparrow (k-2)]\}}} \end{aligned}$$

and

$$[(n+1) \uparrow\uparrow (k-2)] + [n \uparrow (n+1) \uparrow\uparrow (k-2)] \leq (n+1) \uparrow\uparrow (k-1)$$

as developing $(n+1) \uparrow (n+1) \uparrow\uparrow (k-2)$ with Pascal's triangle one obtains, among the others, the term $[(n+1) \uparrow\uparrow (k-2)] \cdot n^1 \cdot 1^{(n+1)\uparrow\uparrow(k-2)-1}$ which is greater than the first term of the addition.

Thus, n in k triangles is always $\leq n^{n^{[(n+1)\uparrow\uparrow(k-1)]}}$.

Both inductions start from the case $k = 1$, for which all three quantities are trivially equal to n^n (using the rules from Remark 2.3). \square

Lemma 4.6. *For all positive integers A, B, C we have*

$$(A \uparrow\uparrow B) \uparrow\uparrow C \leq A \uparrow\uparrow (B + C).$$

Proof. We start by excluding the trivial cases $A = 1 \vee B = 1$. We will use the abbreviation $E := A \uparrow\uparrow (B - 1)$.

We prove more specifically that

$$(A \uparrow\uparrow B) \uparrow\uparrow C \leq A \uparrow A \uparrow (A + 1) \uparrow\uparrow (B + C - 3).$$

The original estimate is then a tower one level higher but replacing all $A + 1$ with A , thus abundantly bigger. We proceed by induction, after checking that the case $C = 1$ is trivial. For the induction step we see immediately that:

$$\begin{aligned} (A \uparrow\uparrow B) \uparrow\uparrow (C + 1) &= A \uparrow [E \times (A \uparrow\uparrow B) \uparrow\uparrow C] \leq \\ &\leq A \uparrow [E \times A \uparrow A \uparrow (A + 1) \uparrow\uparrow (B + C - 3)] = \\ &= A \uparrow A \uparrow [A \uparrow\uparrow (B - 2) + A \uparrow (A + 1) \uparrow\uparrow (B + C - 3)] \end{aligned}$$

so the thesis follows from the following elementary inequality:

$$\begin{aligned} A \uparrow\uparrow (B - 2) + A \uparrow (A + 1) \uparrow\uparrow (B + C - 3) &\leq (A + 1) \uparrow (A + 1) \uparrow\uparrow (B + C - 3) = \\ &= (A + 1) \uparrow\uparrow (B + C - 2). \end{aligned}$$

\square

Proposition 4.7. *The circle symbol (see next section for the case of **Mega**) is comparable to the pentation in the following way:*

$$n \uparrow\uparrow\uparrow (n + 1) \leq \bigcirc(n) \leq n \uparrow\uparrow (n + 2) \uparrow\uparrow\uparrow n.$$

More precisely, a sequence of k squares has the property:

$$n \uparrow\uparrow\uparrow (k + 1) \leq \square^k(n) \leq n \uparrow\uparrow (n + 2) \uparrow\uparrow\uparrow k.$$

Proof. As before, the first inequality is straightforward (using induction) by Proposition 4.5:

$$\square(n \uparrow\uparrow\uparrow k) \geq (n \uparrow\uparrow\uparrow k) \uparrow\uparrow (1 + n \uparrow\uparrow\uparrow k) \geq n \uparrow\uparrow (n \uparrow\uparrow\uparrow k) = n \uparrow\uparrow\uparrow (k + 1).$$

We point out that both inequalities when $k = 0$ become equalities (using the rules from Remark 2.3), and this provides the induction base for the first inequality.

The second inequality can be proved by induction on $k \geq 1$ using Proposition 4.5 and Lemma 4.6:

$$\begin{aligned} \square(n \uparrow\uparrow (n + 2) \uparrow\uparrow\uparrow k) &\leq [n \uparrow\uparrow (n + 2) \uparrow\uparrow\uparrow k] \uparrow\uparrow [2 + n \uparrow\uparrow (n + 2) \uparrow\uparrow\uparrow k] \leq \\ &\leq n \uparrow\uparrow [2 + (n + 2) \uparrow\uparrow\uparrow k + n \uparrow\uparrow (n + 2) \uparrow\uparrow\uparrow k] \leq \\ &\leq n \uparrow\uparrow (n + 2) \uparrow\uparrow\uparrow (k + 1). \end{aligned}$$

It remains to verify the second inequality in the case $k = 1$ which provides the inductive base, but this is immediate. \square

Lemma 4.8. *When $k \geq 2$ one has:*

$$(A \uparrow^k B) \uparrow^k C \leq A \uparrow^k (B + C).$$

Proof. Let $A \geq 3$ and $B \geq 2$. We proceed by induction on k , remarking that Lemma 4.6 gives the starting case $k = 2$, thus supposing that the assertion holds already for $k - 1 \geq 2$. We will use the abbreviation $E := A \uparrow^k (B - 1)$.

We prove more specifically that:

$$(A \uparrow^k B) \uparrow^k C \leq A \uparrow^{k-1} (A + 1) \uparrow^k (B + C - 2).$$

The original estimate is then a tower of \uparrow^{k-1} -hyperoperations one level higher but replacing all $A + 1$ with A , thus abundantly bigger. We now proceed by induction on C , after checking that the case $C = 1$ is trivial. For the induction step we see immediately that:

$$\begin{aligned} (A \uparrow^k B) \uparrow^k (C + 1) &= (A \uparrow^{k-1} E) \uparrow^{k-1} [(A \uparrow^k B) \uparrow^k C] \leq \\ &\leq A \uparrow^{k-1} [E + (A \uparrow^k B) \uparrow^k C] \leq \\ &\leq A \uparrow^{k-1} [E + A \uparrow^{k-1} (A + 1) \uparrow^k (B + C - 2)] \leq \\ &\leq A \uparrow^{k-1} [(A + 1) \uparrow^k (B + C - 1)] \end{aligned}$$

as expected.

The case $A = 1 \vee B = 1$ is trivial, and the easy case $A = 2 \wedge B \geq 2$ is left to the reader. \square

Theorem 4.9. *The Steinhaus-Moser generalized function is comparable to Knuth's up-arrow notation in the following way, for all $n \geq 1$ and $m \geq 2$:*

$$n \uparrow^m (n+1) \leq SM_{m+2}(n) \leq n \uparrow^{m-1} (n+m-1) \uparrow^m n < (n+m-1) \uparrow^m (n+1).$$

More precisely, for all $n \geq 1$, $m \geq 2$ and $k \geq 0$, we have

$$n \uparrow^m (k+1) \leq SM_{m+1}^k(n) \leq n \uparrow^{m-1} (n+m-1) \uparrow^m k.$$

Proof. We point out again that both inequalities when $k = 0$ become trivial equalities (using the rules from Remark 2.3). As before, with a straightforward double induction (m/k) we can prove the left inequality:

$$\begin{aligned} SM_{m+1}^k(n) &= SM_{m+1}(SM_{m+1}^{k-1}(n)) \geq \\ &\geq SM_{m+1}^{k-1}(n) \uparrow^{m-1} (SM_{m+1}^{k-1}(n) + 1) \geq \\ &\geq [n \uparrow^m k] \uparrow^{m-1} [n \uparrow^m k] \geq \\ &\geq n \uparrow^{m-1} [n \uparrow^m k] = n \uparrow^m (k+1). \end{aligned}$$

The right inequality can be proved again by double induction. Using the inductive hypothesis for $m-1$ we get $SM_{m+1}(n') \leq n' \uparrow^{m-1} (n'+m-1)$, and if $k \geq 1$, with Lemma 4.8, we have

$$\begin{aligned} SM_{m+1}^{k+1}(n) &= SM_{m+1}(SM_{m+1}^k(n)) \leq \\ &\leq SM_{m+1}^k(n) \uparrow^{m-1} [m-1 + SM_{m+1}^k(n)] \leq \\ &\leq [n \uparrow^{m-1} (n+m-1) \uparrow^m k] \uparrow^{m-1} [m-1 \\ &\quad + n \uparrow^{m-1} (n+m-1) \uparrow^m k] \leq \\ &\leq n \uparrow^{m-1} [m-1 + (n+m-1) \uparrow^m k + n \uparrow^{m-1} (n+m-1) \uparrow^m k] \leq \\ &\leq n \uparrow^{m-1} [(n+m-1) \uparrow^{m-1} (n+m-1) \uparrow^m k] = \\ &= n \uparrow^{m-1} (n+m-1) \uparrow^m (k+1). \end{aligned}$$

□

4.3.1. Examples: Mega and Megiston. Proposition 4.5 lets us have bounds for the number **Mega** as follows:

$$256 \uparrow\uparrow 257 \leq \textcircled{2} \leq 256 \uparrow 256 \uparrow 257 \uparrow\uparrow 255 \leq 257 \uparrow\uparrow 257$$

because, as we have seen before, it can also be expressed as $\square(256)$.

The **Megiston** is instead approximable at pentation level by Proposition 4.7 with the following bounds:

$$10 \uparrow\uparrow\uparrow 11 \leq \textcircled{10} \leq 10 \uparrow\uparrow 12 \uparrow\uparrow\uparrow 10 < 12 \uparrow\uparrow\uparrow 11.$$

Acknowledgement. The corresponding author is indebted to Dr. Andrea Melicchio and Dr. Enrico Zanardo for their work of reading and editing the original manuscript, for updating bibliographic information and, above all, for their invaluable assistance during the entire publication process.

References

- [1] W. Ackermann, *Zum Hilbertschen Aufbau der reellen Zahlen*, Math. Ann., 99(1) (1928), 118-133.
- [2] L. Antoniotti, F. Caldarola, G. d'Atri and M. Pellegrini, *New approaches to basic calculus: an experimentation via numerical computation*, In: Y.D. Sergeyev and D. Kvasov (eds.) Numerical Computations: Theory and Algorithms NUMTA 2019, Lecture Notes in Computer Science, vol. 11973 (2020), Springer, Cham, 329-342.
- [3] L. Antoniotti, F. Caldarola and M. Maiolo, *Infinite numerical computing applied to Hilbert's, Peano's, and Moore's curves*, Mediterr. J. Math., 17(3) (2020), 99 (19 pp).
- [4] G. d'Atri, *Logic-based consistency checking of XBRL instances*, IJACT 3 (2014), 126-131.
- [5] A. A. Bennett, *Note on an Operation of the Third Grade*, Ann. of Math, Second Series, 17(2) (1915), 74-75.
- [6] J. Bowers, *Exploding Array Function*, Retrieved on 21-11-2018, <http://www.polytope.net/hedrondude/array.htm>.
- [7] F. Caldarola, *The exact measures of the Sierpiński d -dimensional tetrahedron in connection with a Diophantine nonlinear system*, Commun. Nonlinear Sci. and Numer. Simul., 63 (2018), 228-238.
- [8] F. Caldarola, *The Sierpinski curve viewed by numerical computations with infinities and infinitesimals*, Appl. Math. Comput., 318 (2018), 321-328.
- [9] F. Caldarola, D. Cortese, G. d'Atri and M. Maiolo, *Paradoxes of the infinite and ontological dilemmas between ancient philosophy and modern mathematical solutions*, In: Y.D. Sergeyev and D. Kvasov (eds.) Numerical Computations: Theory and Algorithms NUMTA 2019, Lecture Notes in Computer Science, vol. 11973 (2020), Springer, Cham, 358-372.
- [10] F. Caldarola, G. d'Atri, M. Maiolo and G. Pirillo, *New algebraic and geometric constructs arising from Fibonacci numbers. In honor of Masami Ito*, Soft Computing, 24(23) (2020), 17497-17508.

- [11] F. Caldarola, G. d'Atri, M. Maiolo and G. Pirillo, *The sequence of Carboncettus octagons*, In: Y.D. Sergeyev and D. Kvasov (eds.) Numerical Computations: Theory and Algorithms NUMTA 2019, Lecture Notes in Computer Science, vol. 11973 (2020), Springer, Cham, 373-380.
- [12] F. Caldarola, G. d'Atri, P. Mercuri and V. Talamanca, *On the arithmetic of Knuth's powers and some computational results about their density*, In: Y.D. Sergeyev and D. Kvasov (eds.) Numerical Computations: Theory and Algorithms NUMTA 2019, Lecture Notes in Computer Science, vol. 11973 (2020), Springer, Cham, 381-388.
- [13] F. Caldarola and M. Maiolo, *On the topological convergence of multi-rule sequences of sets and fractal patterns*, Soft Computing, 24(23) (2020), 17737-17749.
- [14] F. Caldarola, M. Maiolo and V. Solferino, *A new approach to the Z-transform through infinite computation*, Commun. Nonlinear Sci. Numer. Simul., 82 (2020), 105019.
- [15] J. W. S. Cassels, *An Introduction to Diophantine Approximation*, Cambridge University Press, New York, 1957.
- [16] J. H. Conway and R. K. Guy, *The Book of Numbers*, Copernicus, New York, 1996.
- [17] R. L. Goodstein, *Transfinite ordinals in recursive number theory*, J. Symbolic Logic, 12 (1947), 123-129.
- [18] R. L. Goodstein, *On the restricted ordinal theorem*, J. Symbolic Logic, 9 (1944), 33-41.
- [19] D. E. Knuth, *Mathematics and computer Science: coping with finiteness*, Science, 194(4271) (1976), 1235-1242.
- [20] A. Leonardis, *Continued fractions in local fields and nested automorphisms*, Ph.D. thesis, 2014.
- [21] R. Munafo, *Inventing New Operators and Functions*, Large Numbers at MROB, Retrieved on 19-11-2019.
- [22] R. Munafo, *Versions of Ackermann's Function*, Large Numbers at MROB, Retrieved on 19-11-2019.
- [23] K. K. Nambiar, *Ackermann Functions and Transfinite Ordinals*, Appl. Math. Lett., 8(6) (1995), 51-53.
- [24] R. Rucker, *Infinity and the Mind: The Science and Philosophy of the Infinite*, Princeton University Press, Princeton, NJ, 2019.

- [25] Y. D. Sergeyev, *Numerical point of view on Calculus for functions assuming finite, infinite and infinitesimal values over finite, infinite and infinitesimal domains*, *Nonlinear Anal.*, 71(12) (2009), 1688-1707.
- [26] H. Steinhaus, *Mathematical Snapshots*, 3rd Edition, Oxford University Press, New York, 1969.

Antonino Leonardis and **Gianfranco d'Atri** (Corresponding Author)

Department of Mathematics and Computer Science

University of Calabria

Cubo 31/A, Ponte Bucci, 86036 Arcavacata di Rende (CS), Italy

e-mails: antonino.leonardis@unical.it (A. Leonardis)

datri@mat.unical.it (G. d'Atri)

Fabio Caldarola

Department of Mathematics and Computer Science

University of Calabria

Cubo 30/B, Ponte Bucci, 86036 Arcavacata di Rende (CS), Italy

e-mail: caldarola@mat.unical.it